Technical Disclosure Commons

Defensive Publications Series

July 2020

COMMON ENCRYPTION/ DECRYPTION SPECIFICATION

Naman Bansal

Pankaj Taneja

TITLE: COMMON ENCRYPTION/ DECRYPTION SPECIFICATION

VISA

Naman Bansal

Pankaj Taneja

TECHNICAL FIELD

[001] The present disclosure relates generally to real-time transactions, and more specifically to method and system of encrypting/ decrypting transaction details using a common specification.

BACKGROUND

[002] Real-time transactions are electronic money transfers made from one person (a sender) to another person (a receiver), for instance between a user and a merchant. The real-time transactions allow users to transfer funds immediately from their bank account to merchant's account. Transaction details such as card number, Primary Account Number (PAN), Personal Identification Number (PIN), etc., are encrypted by a payment gateway before being processed by a network processor. Generally, the payment cards (Europay^{R} Mastercard^{R} Visa^{R} (EMV)/ Magnetic Strip Reader (MSR), PIN-debit, PIN-credit) comprises data associated with a plurality of EMV tags. The EMV tags, for example, Tag 56 is associated with Track 1 data, Tag 57 is associated with Track 2 data and the like.

[003] Conventionally, merchant specifies new EMV tags according to need basis, and the payment gateway needs to be reprogrammed according to merchant requirements. Hence, there exists a technical challenge for the payment gateway to reprogram the new tags to meet the merchant requirements. Therefore, there is a need to develop a standard specification for the new tags, such that the common specification can be shared with the merchants, and the merchants update their systems according to the common specification.

[004] The information disclosed in this background of the disclosure section is only for enhancement of understanding of the general background of the invention and should not be taken as an acknowledgement or any form of suggestion that this information forms the prior art already known to a person skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

The example embodiment(s) of the present invention are illustrated by way of example, and not in way by limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[005] Figure 1 illustrates an exemplary platform for performing transactions based on common specification for encrypting and decrypting EMV tags, in accordance with an embodiment of the present disclosure;

[006] Figure 2 shows a flow chart illustrating a method of encrypting EMV tags according to a common specification, in accordance with an embodiment of the present disclosure; and

[007] Figure 3 shows a block diagram of a general-purpose computer system for performing transactions based on common specification for encrypting and decrypting EMV tags, in accordance with embodiments of the present disclosure.

[008] It should be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative systems embodying the principles of the present subject matter. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudo code, and the like represent various processes which may be substantially represented in

computer readable medium and executed by a computer or processor, whether or not such computer or processor is explicitly shown. While each of the figures illustrates a particular embodiment for purposes of illustrating a clear example, other embodiments may omit, add to, reorder, and/or modify any of the elements shown in the figures.

DETAILED DESCRIPTION

[009] In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

[0010] While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the scope of the disclosure.

[0011] The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the system or apparatus.

[0012] Embodiments of the present disclosure relates to encrypting/ decrypting EMV tags using a common specification.

[0013] Figure 1 describes an overview of a platform (100) for performing transactions. The platform (100) is provided as a service to a merchant (101) and a user (105) to enable the merchant (101) and the user (105) to perform transactions. The present disclosure pertains to real-time transactions. The platform (100) comprises a merchant server (102), an acquirer bank (106), a computer system (103), an interoperability server (104) (also referred as network processor) and an issuer bank (107). The merchant (101) may be a person receiving an amount from the user (105). The merchant server (102) is an entity associated with the merchant (101) that provides payment services to the merchant (101). The merchant server (102) is associated with the acquirer bank (106). The acquirer bank (106) is a financial institution that processes credit or debit card payments on behalf of the merchant (101). The payment gateway (103) may receive a payment request message from the merchant server (102) for the transaction between the user (105) and the merchant (101). The payment gateway (103) may encrypt the transaction details received from the merchant server (102) and generate an encrypted payment request message. The payment gateway (103) further provides the encrypted payment request message to the network processor (104). For example, the payment gateway may be an institution such as Cybersource^{R}. The network processor (104) may process transactions routed to the issuer bank (107) to request authentication/ authorization. In an embodiment, the network processor (104) may facilitate the transaction between various issuer banks and acquirer banks. For example, the network processor (104) may be managed by institutions such as Visa^{R}. The issuer bank (107) provides banking services to the user (105), allowing the user (105) to initiate purchases using different payment options.

[0014] In an embodiment, the payment gateway (103) provides the merchant server (102) with an Application Program Interface (API) to integrate Point Of Sale (POS) device to work with the

payment gateway (103). For example, the POS devices may include card readers for detecting payment cards such as Europay Mastercard Visa (EMV) cards, Magnetic Strip Reader (MSR) cards, and PIN-debit cards, PIN-credit cards, contactless cards, and keyed-in cards. In an embodiment, the POS devices may share a plurality of EMV tags along with the transaction details with the payment gateway (103) to be encrypted. However, the different POS devices include the EMV tags according to the card read by the POS devices. Hence, the payment gateway (103) is required to be programmed according to POS devices of different merchants (101).

[0015] Figure 2 shows a flow chart illustrating a method of encrypting EMV tags according to a common specification, in accordance with some embodiments of the present disclosure. As illustrated in Figure 2, the method (200) may comprise one or more steps. The method (200) may be described in the general context of computer executable instructions. Generally, computer executable instructions can include routines, programs, objects, components, data structures, procedures, modules, and functions, which perform particular functions or implement particular abstract data types.

[0016] The order in which the method (200) is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method. Additionally, individual blocks may be deleted from the methods without departing from the spirit and scope of the subject matter described herein. Furthermore, the method can be implemented in any suitable hardware, software, firmware, or combination thereof.

[0017] At step (201), the payment gateway (103) receives an input blob from the merchant server (102). When the user (101) initiates a transaction by providing card details to the POS device (by inserting the EMV card, or by swiping the MSR card, or by keying in the card details), the POS device sends the transaction details along with the card details to the payment gateway (103).

[0018] In an embodiment, during testing the integration of a POS device with the payment gateway (103), the POS may share a Base Derivation Key (BDK) and Key Serial Identification (KSI) including Issuer Identification Number (IIN), Card Identification Number (CID) and Gateway Identification Number (GID) to the payment gateway (103). Once the testing is complete, a new BDK and KSI are generated for real-time transactions. Generally, BDK is known only to POS device manufacturer and the merchant (101). Further identification keys are generated based on the BDK and the BDK itself is not shared. Device identifier is shared in Hex format. The device identifier helps decryption service API to identify source of the payment data. The decryption service supports different types of transactions including reader, tokenized, Visa checkout, keyed-in, etc.

FID(Device Identifier)	Value
ASCII	FID=EMV.PAYMENT.API
HEX	4649443D454D562E5041594D454E542E4150495049

Table 1

[0019] An example of data BDK and KSI are given below in Table 2.

Test BDK (full)	0123ABCFABABCDEFFEDCBA9876543210
Test KSN (Sample KSN)	12345601010018400013
IIN	123456
CID	01
GID	01

Table 2

[0020] An example of PIN BDK and KSI are given below in Table 3.

Test BDK (full)	01234CB392807A6457D3B35D38EAA142
Test KSN (Sample KSN)	12345601020000200004
IIN	123456
CID	01
GID	02

Table 3

[0021] The encryption scheme used is provided in Table 4.

Key Management	DUKPT
Encryption	3DES, CBC, IV=0
Track 1 Padding	Zero Padding
Track 2 Padding	PKCS-7
PAN(ONLY KEYED-IN)	PKCS-7 append char 'F' in case of odd number CC number before padding

Table 4

[0022] Below Table 5 shows an example of PIN-block input format:

Format 0	ISO 9564-1 Format 0, also referred to an ANSI, encrypted with a key derived using DUKPT	API VALUE: 07
Format 1	ISO 9564-1 Format 1 encrypted with a key derived using DUKPT	API VALUE: 08

Table 5

[0023] Below Table 6 shows an example of PIN-block output format:

Format 0	0 Convert to ANSI	API VALUE: 00
Format 1	Retain Input Block Format	API VALUE: 01

Table 6

[0024] In an embodiment, the below example shows the data for MSR samples:

Clear text Track1 Sample:

254233373339353331393233531303034 5E5354414E444152442049534F20202020202020202020202020205E3230303131303131353032313233343 53F230

Clear text Track2 Sample:

3B3337333935333139323335313030343D32303031313031313530323132333435303030303 03F3C

Clear text Track1 Sample with Padding:

254234373136393737383535383437 5E464E414D454C4E414D4520202020202020202020202020205E3230303331303131353032313233343 53F6A00000000000000000000000000000000000000000000000

Clear text Track2 Sample with Padding:

4716977855847=25121011499178 7?004040404

[0025] In an embodiment, the below example shows the data for chip/ contactless card reader samples:

Clear text Track2 Sample with Padding:

5128570128899976D20096220000141F060606060606

[0026] In an embodiment, the below example shows the data for keyed-in card reader samples:

Clear text PAN:

4007000000027F01

[0027] In an embodiment, the below example shows the data for PIN samples:

Format-0

PIN: 1234

PAN: 4111111111111111

Encoded PIN block before encryption: 041225EEEEEEEEEE

Format-1

PIN: 1234

Encoded PIN block before encryption: 141234FFFFFFFFFF

[0028] In an embodiment the input blob should be sent to payment gateway (103) in Tag Length Value (TLV) format. The payment gateway (103) accepts all the standard EMV tags. In an embodiment, length of the value is mentioned in HEX format. Below is one example of TLV format.

5F20104341524420352F564953412054455354

[0029] In an embodiment, the first two bytes of the TLV formatted data indicates the tag ID in HEX format, the next two bytes indicates the length of the value in HEX and the subsequent bytes indicates the data value in ASCII.

[0030] In an embodiment, the EMV tags are divided into three parts: Encrypted Data Tags, Clear Text Tags and Extra data.

[0031] In an embodiment, for EMV/ chip / contactless tags, only track 2 data should be encrypted and rest of the tags should be in clear text format. Currently track 1 and track 3 data are not supported in the EMV/ chip/ contactless tags.

[0032] At step (202), the payment gateway (103) places the input blob in the required EMV tag. For example, if the input track 2 data is: 5128570128899976D20096220000141F, the payment gateway (103) places the input track 2 data in the tag 57. The resulting string is: 57105128570128899976D20096220000141F.

[0033] At step (203), the payment gateway converts the input blob placed in the required EMV tag into the TLV format. For the above track 2 data placed in the tag 57, the TLV formatted data stream is: 5718E9F78AC3DD065F5F76E39F422FC448FB411D406E25004FD2.

[0034] In an embodiment, for the MSR, the payment gateway (103) accepts track 1 and track 2. The track 1 data is sent in tag D1 and track 2 data is sent in tag D2. The TLV formatted data for track 1 and track 2 are given below:

Track 1:
D140704F46D8598D8DD8DA7DCF26BFBA0BAA407662C802CDDAC8321289665226B410AA2177C8D332D202E111D78EE7787B21061B38D1E65C7D7702963F7240946A1

Track 2:
D2283F368E9D161FAF7ED3FB2B24E97CD9D4CA7F32D268D2FE7974E0028CDA0E19064731C1D20C8BE376D40F9F33036028C89F3901805F30020201

[0035] In an embodiment, for the keyed-in transactions, the PAN is sent in encrypted payload, and the PAN data is sent in tag 5A. The TLV formatted data for keyed-in transaction is 5A08BDF1B766347BD297.

[0036] In an embodiment, for the PIN transactions, the PIN is sent in encrypted payload, and the PAN data is sent in tag D6. The TLV formatted data for PIN transaction is D608BB5E95C336F85F99.

[0037] In an embodiment, custom tags are defined to identify the types of transactions in the Table 7:

TAG	**Description**	**Type**	**Attribute**
DFEE12	EMV/CHIP/CONTACT LESS KSN tag. This tag helps identify that it's a CHIP/CONTACT less transaction.	HEX	Mandatory for CHIP/CONTACT-LESS transactions

D0	MSR/Swipe/Track KSN tag. This tag helps identify that it's a MSR/SWIPE transaction.	HEX	Mandatory for MSR transactions
D1	Track1 data tag	HEX	Mandatory for MSR transactions in case there is no track2
D2	Track 2 data tag	HEX	Mandatory for MSR transactions in case there is no track1
D4	Extra tags	HEX	Optional
D5	KEYED-IN KSN tag. This tag helps identify that it's a KEYED-IN transaction.	HEX(10 bytes)	Mandatory for KEYED-IN transactions
D3	PIN KSN tag. This tag helps identify that it's a PIN Based transaction.	HEX(10 bytes)	Mandatory for PIN-BASED transactions
D6	Encrypted PIN Block Tag	HEX(8 bytes)	Mandatory for PIN-BASED transactions

Table 7

[0038] An example of encrypted data and decrypted data is given below for MSR transaction:

Encrypted data:
D00A6299490100000000025D140704f46d8598d8dd8da7dcf26bfba0baa407662c802cddac832
1289665226b410aa2177c8d332d202e111d78ee7787b2106138d1e65c7d7702963f72409e46a1D
2283f368e9d161faf7ed3fb2b24e97cd9d4ca7f32d268d2fe7974e0028cda0e19064731c1d20c8be3
76D40F9F33036028C89F3901805F30020201

Decrypted data:

D0 (?)

6299490100000000025

D1 (?)

704F46D8598D8DD8DA7DCF26BFBA0BAA407662C802CDDAC8321289665226B410AA21
77C8D332D202E111D78EE7787B21061B38D1E65C7D7702963F72409E46A

D2 (?)

3F368E9D161FAF7ED3FB2B24E97CD9D4CA7F32D268D2FE7974E0028CDA0E19064731C
1D20C8BE376

D4 (?)

9F33036028C89F3901805F30020201

[0039] An example of encrypted data and decrypted data is given below for EMV/ chip/ contactless transaction:

Encrypted data:

DFEE120A629949010000000000045718E9F78AC3DD065F5F76E39F422FC448FB411D406E
25004FD24F07A0000000031010950580000080009A031707289B026009C01005F2010434152
4420352F5649534120544553545F24031712319F02060000000039009F03060000000000009F1
A0208409F1C083131323233333343492608F0F6C1FD58D6A5449F2701809F34031E0300

Decrypted data:

DFEE12 (?) 6299490100000000004

57 (track 2 equivalent data) E9F78AC3DD065F5F76E39F422FC448FB411D406E25004FD2

4F (ADF - Application dedicated file name) A0000000031010

95 (TVR - Terminal Verification Results)

[0040] In an embodiment, errors can occur in the transaction data. For example, all bits zeroes is a clean bill of health and means the transaction could be approved offline (without contacting the issuer). Any non-zero bit could cause a decline or a need to contact the issuer (go online). Each bit is a single fact about the transaction. Below are further examples:

8000000000 (Byte 1 Bit 8) Offline data authentication was not performed

0000008000 (Byte 4 Bit 8) Transaction exceeds floor limit

9A (transaction date) 170728

9B (TSI - Transaction Status Indicator)

4000 (Byte 1 Bit 7) Cardholder verification was performed

2000 (Byte 1 Bit 6) Card risk management was performed

0800 (Byte 1 Bit 4) Terminal risk management was performed

9C (transaction type) 00

5F20 (card holder name) CARD 5/VISA TEST

5F24 (card expiry) 171231

9F02 (amount authorized) 000000003900

9F03 (amount other) 000000000000

9F1A (terminal country code) USA (United States)

9F1C (terminal id) 11223344

9F26 (application cryptogram) F0F6C1FD58D6A544

9F27 (cryptogram information data) ARQC (Authorization Request Cryptogram - Go ask the issuer)

9F34 (CVM Results - Cardholder Verification Results)

1E Signature

03 If terminal supports CVM

00 Unknown

[0041] An example of encrypted data and decrypted data is given below for keyed-in transaction:

Encrypted data:
D50A6299490800001F4000985A08BDF1B766347BD297

Decrypted data:

D5 (?) 6299490800001F400098

5A (PAN) BDF1B766347BD297

[0042] An example of encrypted data and decrypted data is given below for PIN-debit transaction:

Encrypted data:
DFEE120A6299490100000000000045718E9F78AC3DD065F5F76E39F422FC448FB411D406E25004FD24F07A0000000031010950580000080009A031707289B026009C01005F2010434152
4420352F5649534120544553545F24031712319F02060000000039009F03060000000000009F1A0208409F1C0831313232333343492608F0F6C1FD58D6A5449F2701809F34031E0300D30AFFFF1B1D140000000005D60852F20658C04DB351D70108D80101D905FFFF000003

Decrypted data:

DFEE12 (?) 88888851400018400013

57 (track 2 equivalent data) 76B59D0808D703A6FE8B91BFE7CFC17FA56E621F4F0F13C8

4F (ADF - Application dedicated file name) A0000000031010

95 (TVR - Terminal Verification Results)

8000000000 (Byte 1 Bit 8) Offline data authentication was not performed

0000008000 (Byte 4 Bit 8) Transaction exceeds floor limit

9A (transaction date) 170728

9B (TSI - Transaction Status Indicator)

A record of things that happened during the transaction.

Whilst the TVR is expected to mainly be zeroes. This field is expected to mainly be ones. Each bit is a fact about the transaction.

4000 (Byte 1 Bit 7) Cardholder verification was performed

2000 (Byte 1 Bit 6) Card risk management was performed

0800 (Byte 1 Bit 4) Terminal risk management was performed

9C (transaction type) 00

5F20 (card holder name) CARD 5/VISA TEST

5F24 (card expiry) 171231

9F02 (amount authorized) 000000003900

9F03 (amount other) 000000000000

9F1A (terminal country code) USA (United States)

9F1C (terminal id) 11223344

9F26 (application cryptogram) F0F6C1FD58D6A544

9F27 (cryptogram information data) ARQC (Authorization Request Cryptogram - Go ask the issuer)

9F34 (CVM Results - Cardholder Verification Results)

1E Signature

03 If terminal supports CVM

00 Unknown

D3 (?) FFFF1B1D140000000005

D6 (?) 52F20658C04DB351

D7 (?) 08

D8 (?) 01

D9 (?) FFFF000003

[0043] An example of a sample track 1 data and response is given below for EMV/ chip/ contactless transaction:

Encrypted data:

DFEE120A629949010000000000045718e9f78ac3dd065f5f76e39f422fc448fb411d406e25004fd24F07A0000000310109505800000800 09A031707289B026009C01005F20104341524420352F5649534120544553545F24031712319F02060000000039009F03060000000000009F1A0208409F1C083131323233333434926 08F0F6C1FD58D6A5449F2701809F34031E0300

Service decrypted response:

<Data>

<Track2>3B34373631333430303030303030303035303D313731323230313132353233343531 3F</Track2>

<PAN>4761340000000050</PAN>

<CardholderName>CARD 5/VISA TEST</CardholderName>

<ExpiryDate>12/17</ExpiryDate>

<ExpiryYear>17</ExpiryYear>

<ExpiryMonth>12</ExpiryMonth>

<EmvData>

<TlvData>5710476134000000050D1712201125234514F07A0000000310109505800000800 09A031707289B0268009C01005F20104341524420352F5649534120544555545F24031712319F02060000000039009F03060000000000009F1A0208409F1C083131323233333434 9F2608F0F6C1FD58D6A5449F2701809F34031E0300</TlvData>

<EmvOnlineMessage>

<EmvTag>

<TagId>4F</TagId>
<TagLen>07</TagLen>
<TagData>A0000000031010</TagData>
</EmvTag>
<EmvTag>
<TagId>95</TagId>
<TagLen>05</TagLen>
<TagData>8000008000</TagData>
</EmvTag>
<EmvTag>
<TagId>9A</TagId>
<TagLen>03</TagLen>
<TagData>170728</TagData>
</EmvTag>
<EmvTag>
<TagId>9B</TagId>
<TagLen>02</TagLen>
<TagData>6800</TagData>
</EmvTag>
<EmvTag>
<TagId>9C</TagId>
<TagLen>01</TagLen>
<TagData>00</TagData>
</EmvTag>
<EmvTag>
<TagId>5F20</TagId>
<TagLen>10</TagLen>
<TagData>4341524420352F564953412054455354</TagData>
</EmvTag>
<EmvTag>
<TagId>5F24</TagId>
<TagLen>03</TagLen>
<TagData>171231</TagData>
</EmvTag>
<EmvTag>

<TagId>9F02</TagId>
<TagLen>06</TagLen>
<TagData>000000003900</TagData>
</EmvTag>
<EmvTag>
<TagId>9F03</TagId>
<TagLen>06</TagLen>
<TagData>000000000000</TagData>
</EmvTag>

[0044] An example of a sample track 1 data, track 2 data and response is given below for MSR transaction:

Encrypted data:

D00A6299490100000000025D140704f46d8598d8dd8da7dcf26bfba0baa407662c802cddac832
1289665226b410aa2177c8d332d202e111d78ee7787b2106138d1e65c7d7702963f72409e46a1D
2283f368e9d161faf7ed3fb2b24e97cd9d4ca7f32d268d2fe7974e0028cda0e19064731c1d20c8be3
76D40F9F33036028C9F3901805F30020201

Service decrypted response:
<Data>
<Track1>2542333733393533313932333531303034 5E5354414E444152442049534F202020202
02020202020202020205E3230303131303131353032313233343 53F</Track1>
<Track2>3B3337333935333139323335313030343D32303031313031313530323132333435303
03030303F</Track2>
<PAN>373953192351004</PAN>
<Surname>STANDARD ISO</Surname>
<Firstname/>
<ExpiryDate>01/20</ExpiryDate>
<ExpiryYear>20</ExpiryYear>
<ExpiryMonth>01</ExpiryMonth>
<EmvExtra>
<TlvData>9F33036028C89F3901805F30020201</TlvData>
<EmvTags>

<EmvTag>
<TagId>9F33</TagId>
<TagLen>03</TagLen>
<TagData>6028C8</TagData>
</EmvTag>
<EmvTag>
<TagId>9F39</TagId>
<TagLen>01</TagLen>
<TagData>80</TagData>
</EmvTag>
<EmvTag>
<TagId>5F30</TagId>
<TagLen>02</TagLen>
<TagData>0201</TagData>
</EmvTag>
</EmvTags>
</EmvExtra>
</Data>

[0045] An example of a sample PAN data and response is given below for manually keyed-in transaction:

Encrypted data:

D50A6299490800001F4000985A08BDF1B766347BD297

Service decrypted response:
<Data>
<PAN>4007000000027</PAN>
</Data>

[0046] An example of a sample PAN data and response is given below for PI-debit/ PIN credit transaction:

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Header>
<RequestHeader xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://mobileApi.visa.com/SEC/Request">
<TransactionID>a750ecef-5add-4cd2-b07a-0a350b951d73</TransactionID>
<RequestID>198ccc67-1a86-4543-8a86-a73ec442a52a</RequestID>
<Version>1.0</Version>
</RequestHeader>
</soap:Header>
<soap:Body>
<ServiceRequest xmlns="http://mobileApi.visa.com/SEC/Request">
<SecRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://mobileApi.visa.com/SEC/Request">
<FormOfPayment>
<Version>3.0</Version>
<Data>
<Encoding>Hex</Encoding>
<Algo>TDES</Algo>
<Scheme>
<DUKPT>
<Op>Decrypt</Op>
<DeviceInfo>
<Description>4649443D454D562E5041594D454E542E415049</Description>
</DeviceInfo>
<EncryptedData>
<Value>DFEE120A888888514000184000135718 76B59D0808D703A6FE8B91BFE7CFC17FA
56E621F4F0F13C84F07A0000000031010950580000080009A031707289B026009C01005F201
04341524420352F5649534120544553545F24031712319F020600
00000039009F03060000000000009F1A0208409F1C0831313232333334349F2608F0F6C1FD5
8D6A5449F2701809F34031E0300D30AFFFF1B1D140000000005D0852F20658C04DB351D7
0108D80101D905FFFF000003</Value>

<DateTimeUTC>0001-01-01T00:00:00</DateTimeUTC>
</EncryptedData>
<Sequence>0</Sequence>
<Mode>Data</Mode>
</DUKPT>
</Scheme>
</Data>
</FormOfPayment>
<Version>2.0</Version>
</SecRequest>
</ServiceRequest>
</soap:Body>
</soap:Envelope>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Header>
<h:ResponseHeader xmlns:h="http://mobileApi.visa.com/SEC/Response"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<h:TransactionID>a750ecef-5add-4cd2-b07a-0a350b951d73</h:TransactionID>
<h:RequestID>198ccc67-1a86-4543-8a86-a73ec442a52a</h:RequestID>
<h:Version>3.0.010918</h:Version>
<h:ProcessingTimestamp>2019-08-19T21:03:20.6207627Z</h:ProcessingTimestamp>
<h:ProcessingDuration>2665.9041</h:ProcessingDuration>
</h:ResponseHeader>
</s:Header>
<s:Body>
<ServiceResponse xmlns="http://mobileApi.visa.com/SEC/Response">
<SecResponse xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<FormOfPayment>
<Version>3.0</Version>
<Data>
<Track2>3B3531323835373031323838393939373633D32303039363232303030303134314631463F<
/Track2>
<PAN>5128570128899976</PAN>
<CardholderName>CARD 5/VISA TEST</CardholderName>
<ExpiryDate>09/20</ExpiryDate>

<ExpiryYear>20</ExpiryYear>
<ExpiryMonth>09</ExpiryMonth>
<EmvData>
<TlvData>5716512857012889976D20096220000141F0606060606064F07A000000031010950580000080009A031707289B0268009C01005F2010434152442035 2F56453412054455354 5F24031712319F02060000000039009F03060000000000009F1A0208409F1C08313132323333343349F2608F0F6C1FD58D6A5449F2701809F3031E0300D30AFFFF1B1D140000000005D60852F20658C04DB351D70108D80101D905FFFF000003</TlvData>
<EmvOnlineMessage>
<EmvTag>
<TagId>4F</TagId>
<TagLen>07</TagLen>
<TagData>A0000000031010</TagData>
</EmvTag>
<EmvTag>
<TagId>95</TagId>
<TagLen>05</TagLen>
<TagData>8000008000</TagData>
</EmvTag>
<EmvTag>
<TagId>9A</TagId>
<TagLen>03</TagLen>
<TagData>170728</TagData>
</EmvTag>
<EmvTag>
<TagId>9B</TagId>
<TagLen>02</TagLen>
<TagData>6800</TagData>
</EmvTag>
<EmvTag>
<TagId>9C</TagId>
<TagLen>01</TagLen>
<TagData>00</TagData>
</EmvTag>
<EmvTag>

<TagId>5F20</TagId>
<TagLen>10</TagLen>
<TagData>4341524420352F564953412054455354</TagData>
</EmvTag>
<EmvTag>
<TagId>5F24</TagId>
<TagLen>03</TagLen>
<TagData>171231</TagData>
</EmvTag>
<EmvTag>
<TagId>9F02</TagId>
<TagLen>06</TagLen>
<TagData>000000003900</TagData>
</EmvTag>
<EmvTag>
<TagId>9F03</TagId>
<TagLen>06</TagLen>
<TagData>000000000000</TagData>
</EmvTag>
<EmvTag>
<TagId>9F1A</TagId>
<TagLen>02</TagLen>
<TagData>0840</TagData>
</EmvTag>
<EmvTag>
<TagId>9F1C</TagId>
<TagLen>08</TagLen>
<TagData>3131323233333434</TagData>
</EmvTag>
<EmvTag>
<TagId>9F26</TagId>
<TagLen>08</TagLen>
<TagData>F0F6C1FD58D6A544</TagData>
</EmvTag>
<EmvTag>

<TagId>9F27</TagId>
<TagLen>01</TagLen>
<TagData>80</TagData>
</EmvTag>
<EmvTag>
<TagId>9F34</TagId>
<TagLen>03</TagLen>
<TagData>1E0300</TagData>
</EmvTag>
<EmvTag>
<TagId>D3</TagId>
<TagLen>0A</TagLen>
<TagData>FFFF1B1D140000000005</TagData>
</EmvTag>
<EmvTag>
<TagId>D6</TagId>
<TagLen>08</TagLen>
<TagData>52F20658C04DB351</TagData>
</EmvTag>
<EmvTag>
<TagId>D7</TagId>
<TagLen>01</TagLen>
<TagData>08</TagData>
</EmvTag>
<EmvTag>
<TagId>D8</TagId>
<TagLen>01</TagLen>
<TagData>01</TagData>
</EmvTag>
<EmvTag>
<TagId>D9</TagId>
<TagLen>05</TagLen>
<TagData>FFFF000003</TagData>
</EmvTag>
</EmvOnlineMessage>

```
</EmvData>
<PinEntry>
<epb>2B730EDA9B1A1B50</epb>
<ksn>FFFF1B1D140000000005</ksn>
</PinEntry>
</Data>
</FormOfPayment>
<Status>
<StatusCodeGroup>OK</StatusCodeGroup>
<StatusCode>0</StatusCode>
<Description>Success</Description>
</Status>
<Version>2.0</Version>
</SecResponse>
</ServiceResponse>
</s:Body>
</s:Envelope>
<EmvTag>
<TagId>9F1A</TagId>
<TagLen>02</TagLen>
<TagData>0840</TagData>
</EmvTag>
<EmvTag>
<TagId>9F1C</TagId>
<TagLen>08</TagLen>
<TagData>3131323233333434</TagData>
</EmvTag>
<EmvTag>
<TagId>9F26</TagId>
<TagLen>08</TagLen>
<TagData>F0F6C1FD58D6A544</TagData>
</EmvTag>
<EmvTag>
<TagId>9F27</TagId>
<TagLen>01</TagLen>
```

<TagData>80</TagData>
</EmvTag>
<EmvTag>
<TagId>9F34</TagId>
<TagLen>03</TagLen>
<TagData>1E0300</TagData>
</EmvTag>
</EmvOnlineMessage>
</EmvData>
</Data>

[0047] This invention helps standardized the EMV payload specification for encrypted payment tags, which will help to the payment industry wherein all the tags will be in a standard format. Currently, payment gateways (103) like Cybersource^{R} has introduced open platform integrations which allow device manufacturers to have their own applications on the terminal for payment processing. But, Cybersource^{R} handles all their transactions which requires Cybersource^{R} to decrypt the EMV/TAP/MSR payload. The common specification makes sure payment gateways (103) need not develop again for these integrations and also makes payment gateways (103) future safe.

[0048] Figure 3 illustrates a block diagram of an exemplary computer system (300) for implementing embodiments consistent with the present disclosure. The computer system (300) may comprise a central processing unit (“CPU” or “processor”) (302). The processor (302) may comprise at least one data processor. The processor (302) may include specialized processing units such as integrated system (bus) controllers, memory management control units, floating point units, graphics processing units, digital signal processing units, etc.

[0049] The processor (302) may be disposed in communication with one or more input/output (I/O) devices (not shown) via I/O interface (301). The I/O interface (301) may employ

communication protocols/methods such as, without limitation, audio, analog, digital, monoaural, RCA, stereo, IEEE-1394, serial bus, universal serial bus (USB), infrared, PS/2, BNC, coaxial, component, composite, digital visual interface (DVI), high-definition multimedia interface (HDMI), Radio Frequency (RF) antennas, S-Video, VGA, IEEE 802.n /b/g/n/x, Bluetooth, cellular (e.g., code-division multiple access (CDMA), high-speed packet access (HSPA+), global system for mobile communications (GSM), long-term evolution (LTE), WiMax, or the like), etc.

[0050] Using the I/O interface (301), the computer system (300) may communicate with one or more I/O devices. For example, the input device (310) may be an antenna, keyboard, mouse, joystick, (infrared) remote control, camera, card reader, fax machine, dongle, biometric reader, microphone, touch screen, touchpad, trackball, stylus, scanner, storage device, transceiver, video device/source, etc. The output device (311) may be a printer, fax machine, video display (e.g., cathode ray tube (CRT), liquid crystal display (LCD), light-emitting diode (LED), plasma, Plasma display panel (PDP), Organic light-emitting diode display (OLED) or the like), audio speaker, etc.

[0051] The processor (302) may be disposed in communication with the communication network (309) via a network interface (303). The network interface (303) may communicate with the communication network (309). The network interface (303) may employ connection protocols including, without limitation, direct connect, Ethernet (e.g., twisted pair 10/100/1000 Base T), transmission control protocol/internet protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc. The communication network (309) may include, without limitation, a direct interconnection, local area network (LAN), wide area network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, etc. The network interface (303) may employ connection protocols include, but not limited to, direct connect, Ethernet (e.g., twisted pair 10/100/1000 Base T), transmission control protocol/internet protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc.

[0052] The communication network (309) includes, but is not limited to, a direct interconnection, an e-commerce network, a peer to peer (P2P) network, local area network (LAN), wide area network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, Wi-Fi and such. The first network and the subsequent network may either be a dedicated network or a shared network, which represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP), etc., to communicate with each other. Further, the first network and the subsequent network may include a variety of network devices, including routers, bridges, servers, computing devices, storage devices, etc.

[0053] In some embodiments, the processor (302) may be disposed in communication with a memory (305) (e.g., RAM, ROM, etc. not shown in Figure 3) via a storage interface (304). The storage interface (304) may connect to memory (305) including, without limitation, memory drives, removable disc drives, etc., employing connection protocols such as serial advanced technology attachment (SATA), Integrated Drive Electronics (IDE), IEEE-1394, Universal Serial Bus (USB), fiber channel, Small Computer Systems Interface (SCSI), etc. The memory drives may further include a drum, magnetic disc drive, magneto-optical drive, optical drive, Redundant Array of Independent Discs (RAID), solid-state memory devices, solid-state drives, etc.

[0054] The memory (305) may store a collection of program or database components, including, without limitation, user interface (306), an operating system (307), web server (308) etc. In some embodiments, computer system (300) may store user/application data , such as, the data, variables, records, etc., as described in this disclosure. Such databases may be implemented as fault-tolerant, relational, scalable, secure databases such as Oracle ® or Sybase®.

[0055] The operating system (307) may facilitate resource management and operation of the computer system (300). Examples of operating systems include, without limitation, APPLE MACINTOSH^{R} OS X, UNIX^{R}, UNIX-like system distributions (E.G., BERKELEY SOFTWARE DISTRIBUTION^{TM} (BSD), FREEBSD^{TM}, NETBSD^{TM}, OPENBSD^{TM}, etc.), LINUX DISTRIBUTIONS^{TM} (E.G., RED HAT^{TM}, UBUNTU^{TM}, KUBUNTU^{TM}, etc.), IBM^{TM} OS/2, MICROSOFT^{TM} WINDOWS^{TM} (XP^{TM}, VISTA^{TM}/7/8, 10 etc.), APPLE^{R} IOS^{TM}, GOOGLE^{R} ANDROID^{TM}, BLACKBERRY^{R} OS, or the like.

[0056] In some embodiments, the computer system (300) may implement a web browser (308) stored program component. The web browser (308) may be a hypertext viewing application, for example MICROSOFT^{R} INTERNET EXPLORER^{TM}, GOOGLE^{R} CHROME^{TM0}, MOZILLA^{R} FIREFOX^{TM}, APPLE^{R} SAFARI^{TM}, etc. Secure web browsing may be provided using Secure Hypertext Transport Protocol (HTTPS), Secure Sockets Layer (SSL), Transport Layer Security (TLS), etc. Web browsers (308) may utilize facilities such as AJAX^{TM}, DHTML^{TM}, ADOBE^{R} FLASH^{TM}, JAVASCRIPT^{TM}, JAVA^{TM}, Application Programming Interfaces (APIs), etc. In some embodiments, the computer system (300) may implement a mail server (not shown in Figure) stored program component. The mail server may be an Internet mail server such as Microsoft Exchange, or the like. The mail server may utilize facilities such as ASP^{TM}, ACTIVEX^{TM}, ANSI^{TM} C++/C#, MICROSOFT^{R}, .NET^{TM}, CGI SCRIPTS^{TM}, JAVA^{TM}, JAVASCRIPT^{TM}, PERL^{TM}, PHP^{TM}, PYTHON^{TM}, WEBOBJECTS^{TM}, etc. The mail server may utilize communication protocols such as Internet Message Access Protocol (IMAP), Messaging Application Programming Interface (MAPI), MICROSOFT^{R} exchange, Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), or the like. In some embodiments, the computer system (300) may implement a mail client stored program component. The mail client (not shown in Figure) may be a mail viewing application, such as APPLE^{R} MAIL^{TM}, MICROSOFT^{R} ENTOURAGE^{TM}, MICROSOFT^{R} OUTLOOK^{TM}, MOZILLA^{R} THUNDERBIRD^{TM}, etc.

[0057] Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer-readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term “computer-readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., be non-transitory. Examples include Random Access Memory (RAM), Read-Only Memory (ROM), volatile memory, non-volatile memory, hard drives, Compact Disk (CD) ROMs, DVDs, flash drives, disks, and any other known physical storage media.

[0058] The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

[0059] The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise.

[0060] The enumerated listing of items does not imply that any or all of the items are mutually exclusive, unless expressly specified otherwise. The terms "a", "an" and "the" mean "one or more", unless expressly specified otherwise.

[0061] A description of an embodiment with several components in communication with each other does not imply that all such components are required. On the contrary a variety of optional components are described to illustrate the wide variety of possible embodiments of the invention.

[0062] When a single device or article is described herein, it will be readily apparent that more than one device/article (whether or not they cooperate) may be used in place of a single device/article. Similarly, where more than one device or article is described herein (whether or not they cooperate), it will be readily apparent that a single device/article may be used in place of the more than one device or article or a different number of devices/articles may be used instead of the shown number of devices or programs. The functionality and/or the features of a device may be alternatively embodied by one or more other devices which are not explicitly described as having such functionality/features. Thus, other embodiments of the invention need not include the device itself.

[0063] The illustrated operations of Figure 2 shows certain events occurring in a certain order. In alternative embodiments, certain operations may be performed in a different order, modified or removed. Moreover, steps may be added to the above described logic and still conform to the described embodiments. Further, operations described herein may occur sequentially or certain operations may be processed in parallel. Yet further, operations may be performed by a single processing unit or by distributed processing units.

[0064] Finally, the language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. It is therefore intended that the scope of the invention be limited not by this detailed description, but rather by any claims that issue on an application based here on. Accordingly, the disclosure of the embodiments of the invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

[0065] While various aspects and embodiments have been disclosed herein, other aspects and embodiments will be apparent to those skilled in the art. The various aspects and embodiments disclosed herein are for purposes of illustration and are not intended to be limiting, with the true scope and spirit being indicated by the following claims.

ABSTRACT

The invention relates to using a common specification to encrypt and decrypt EMV tags. Currently, several EMV tags are not included and hence, payment gateways need to provide service based on different merchants. Hence, a common specification solves the problem of the payment gateways re-programming according to the needs to different merchants. Each merchant can program respective POS devices according to the common specification, thus reducing the complexity and burden on the payment gateways.

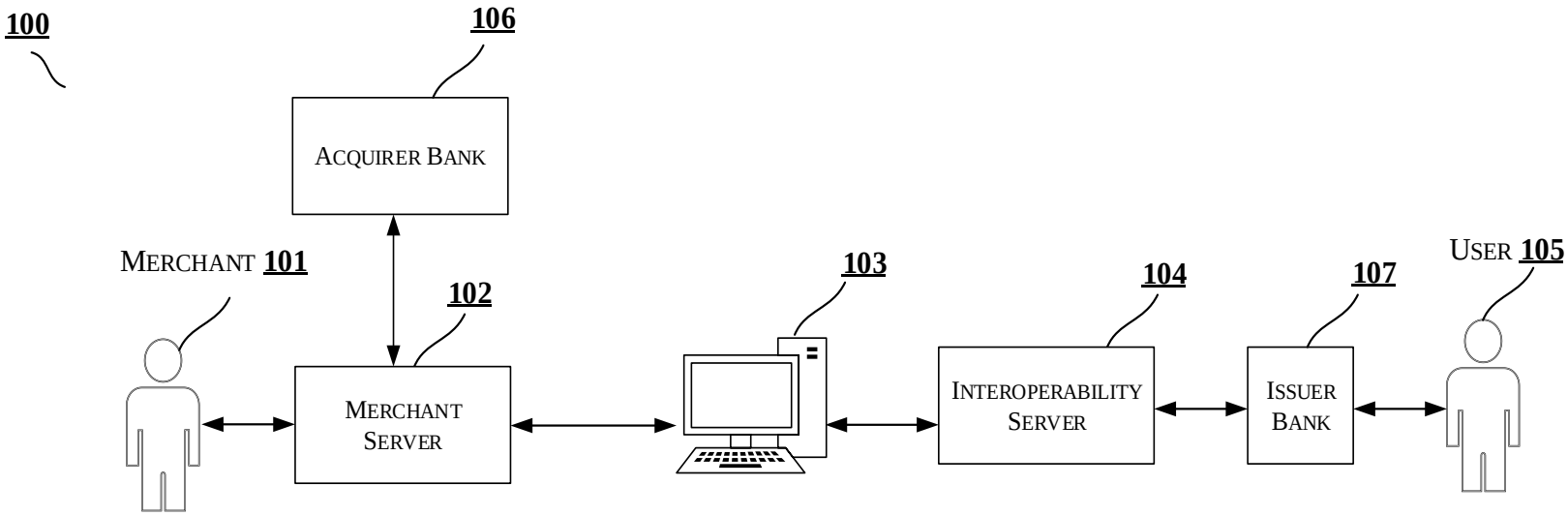
100
106
ACQUIRER BANK
MERCHANT 101
102
103
104
107
USER 105
MERCHANT SERVER
INTEROPERABILITY SERVER
ISSUER BANK

FIGURE 1

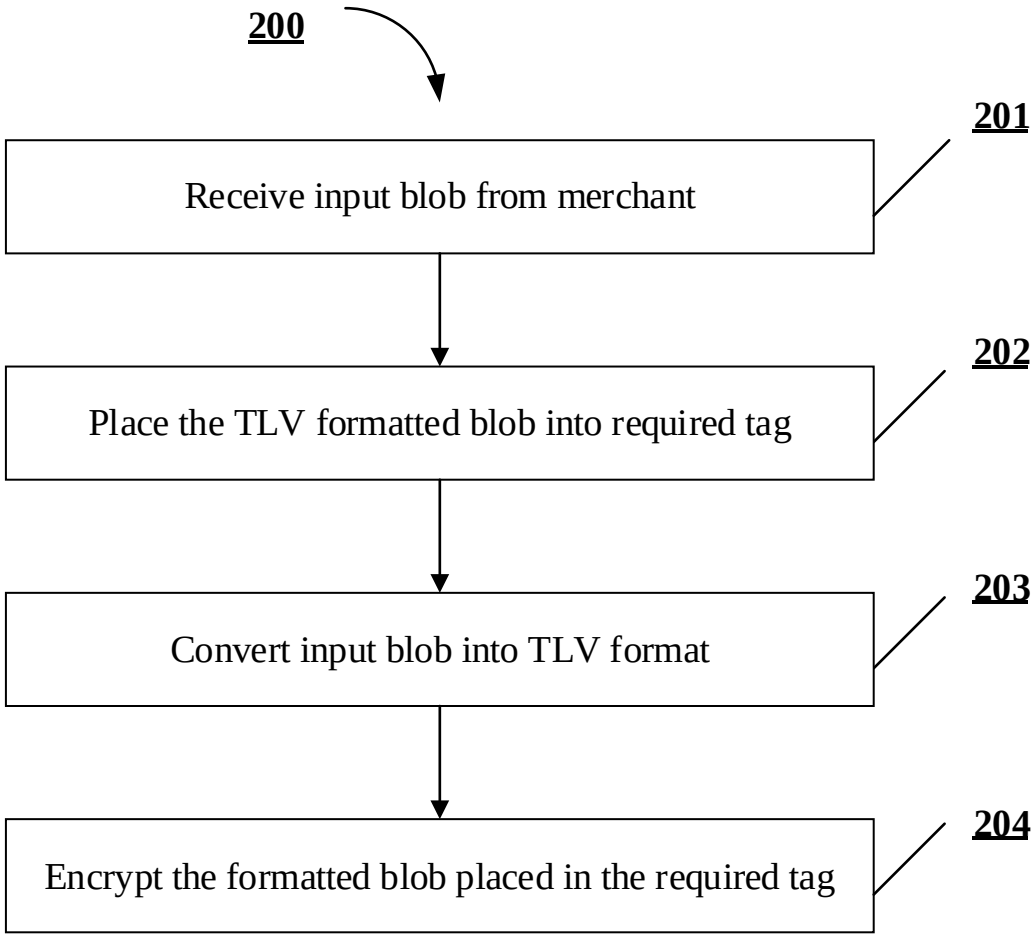
200
201
Receive input blob from merchant
202
Place the TLV formatted blob into required tag
203
Convert input blob into TLV format
204
Encrypt the formatted blob placed in the required tag

FIGURE 2

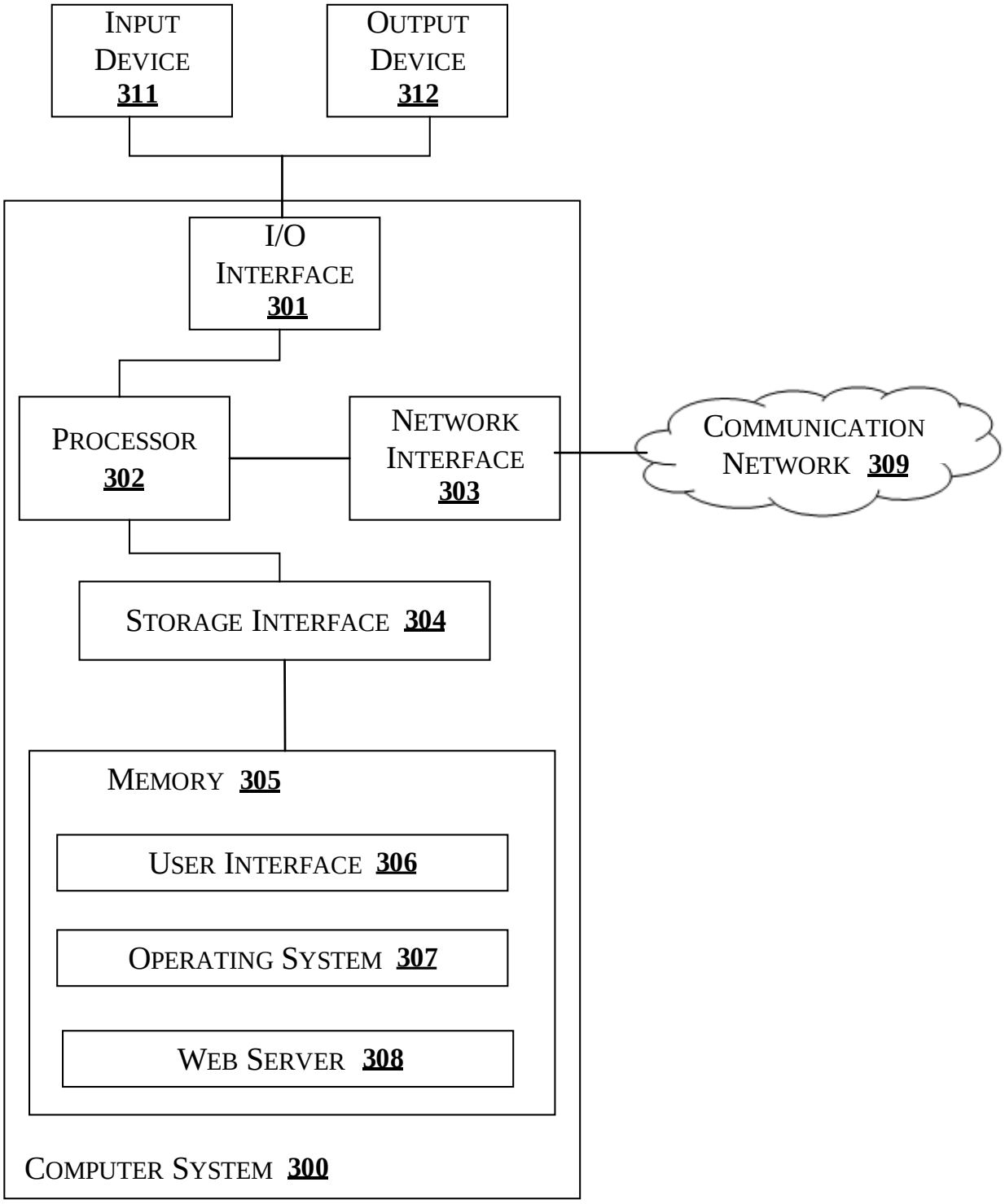
INPUT DEVICE 311
OUTPUT DEVICE 312
I/O INTERFACE 301
PROCESSOR 302
NETWORK INTERFACE 303
COMMUNICATION NETWORK 309
STORAGE INTERFACE 304
MEMORY 305
USER INTERFACE 306
OPERATING SYSTEM 307
WEB SERVER 308
COMPUTER SYSTEM 300

FIGURE 3